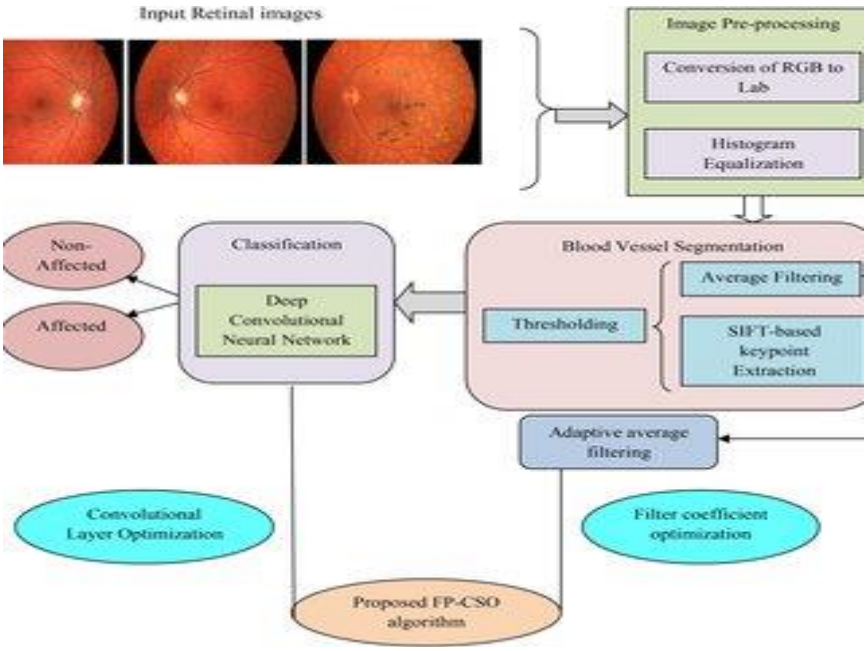
Project Planning Phase

Technology Stack

Date	27th October 2023
Team Id	Team-592538
Project Name	AUTOMATED PREDICTION MODEL FOR DIABETIC RETINOPATHY USING CNN
Maximum Marks	4
Team Members	Sai Krishna Ravulapalli Lokesh.B Varun Kumar Sadineni

Technical Architecture

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table.



Basic Phases of Technical Architecture:

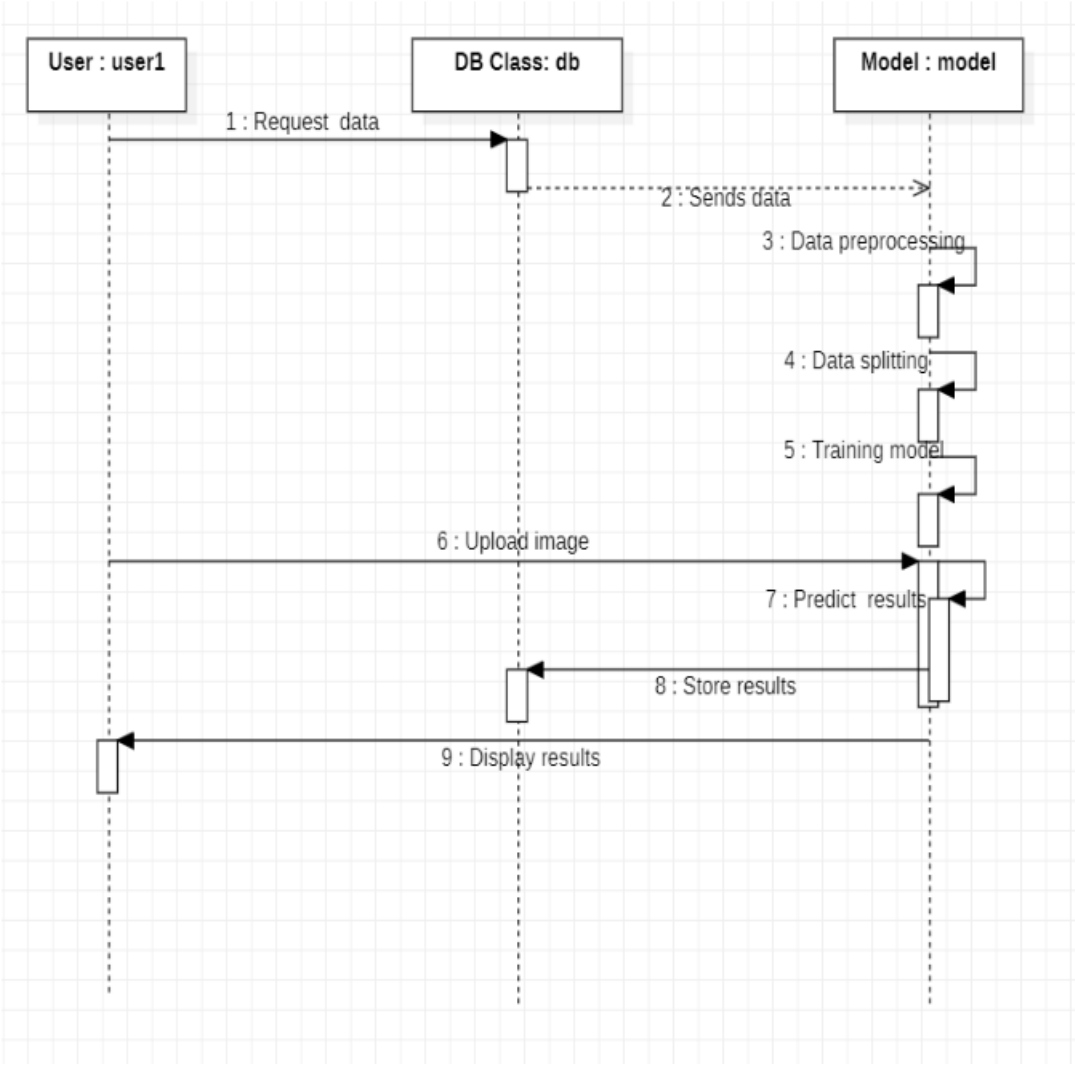


Table 1: Components & Technologies

Component	Description	Technologies
User Interaction	Interface for user interaction with the application along with creating an user friendly interface	Web-based UI (HTML, CSS, JavaScript) or StreamLit platform
Application Logic-1	Core logic responsible for handling user requests	Python, Flask,
Data Collection	Data collection encompasses sourcing historical Bitcoin price data from exchanges via APIs, crucial for Time Series Analysis using Prophet	Integrate Python, Prophet, and cloud infrastructure for accurate predictions.
Data Input	Handling and processing user-provided input data	Forms, file uploads, APIs, or command-line input.
External API	Integration with external data sources or services.	RESTFUL APIs(Bitcoin API, Binance API)
Cloud Database	Storage and management of structured da	Amazon RDS, Google Cloud SQL, or Azure SQL.
Model Integration	Interface for integrating with FbProphet Model	RESTful API endpoints (Flask, FastAPI,JSON,XML
API Model Deployment	Responsible for deploying machine learning models as APIs, enabling real-time predictions and external interaction. It ensures model accessibility and scalability	Docker, Kubernetes, or serverless (AWS Lambda).
Machine Learning Model	The predictive model using Fb Prophet for bitcoin forecasting	Scikit-Learn, Keras TensorFlow, PyTorch, XGBoost, fb Prophet

Data Preprocessing	Data preparation and feature engineering.	Pandas, NumPy, scikit-learn, or custom scripts.
Model Deployment	Hosting and serving the machine learning model.	Flask, FastAPI, TensorFlow Serving, or StreamLit.
Infrastructure (Server/Cloud)	Underlying cloud infrastructure and resources	AWS, Google Cloud, Azure, or on-premises servers like Local, Cloud Foundry, Kubernetes etc.

Table 2: Application Characteristics:

Component	Description	Technologies
Open-Source Frameworks	Leveraging open-source frameworks for model development and deployment guarantees cost-efficiency and flexibility, streamlining user access to Bitcoin price predictions, especially during market fluctuations and high volatility periods, enhancing accuracy and reliability in the Time Series Analysis project for Bitcoin Price Prediction using Prophet.	- Scikit-Learn, TensorFlow, PyTorch for model development. - Flask or FastAPI for API deployment. - Kubernetes for container orchestration. - Jupyter Notebook for model prototyping and development
Security Implementations	Implementations include robust encryption, authentication, and access controls to safeguard sensitive data in the Time Series Analysis project for Bitcoin Price Prediction using Prophet, leveraging Python, Prophet, and cloud-	- OAuth 2.0 or JWT for user authentication. - Encryption (HTTPS/SSL) for data in transit. - Role-based access control. - Regular security audits and updates. - Compliance with industry standards (e.g., GDPR)

	based security protocols for advanced protection.	
Scalable Architecture	Designing a scalable architecture that can handle growing data volumes and user demands which can manage the huge inflow of user demands assuming as a big data	- Microservices architecture for modularity and scalability. - Containerization with Docker and orchestration with Kubernetes. - Load balancers for distributing traffic. - Auto-scaling based on resource usage.
Availability	Ensuring high availability and minimal downtime for the Time Series Analysis application is vital, enabling continuous data processing and accurate Bitcoin price predictions using Prophet for optimal financial insights.	Redundancy in database and API deployment. - Geographically distributed data centers or cloud regions. - Monitoring and alerting systems (e.g., Prometheus, Grafana). - Failover mechanisms for fault tolerance.
Performance	Optimizing application performance to provide quick insights and predictions	Caching mechanisms for frequently accessed data. - Model optimization (e.g., quantization) for faster inference. - Load testing and performance tuning
User-Friendly Interface	Creating an intuitive and user friendly interface for data input, visualization, and interact	HTML, CSS, JavaScript for web-based UI. - React or similar frameworks for responsive design. - Data visualization libraries (e.g., D3.js). - User experience (UX) testing and design principles.

Interoperability and Accuracy	Ensuring seamless integration with external systems and maintaining high prediction accuracy	- RESTful API design for interoperability. - Integration with external data sources (e.g., weather data). - Continuous model monitoring and retraining for accuracy improvement. - Data preprocessing techniques to enhance model accuracy
Data Transparency	Increasing transparency in Bitcoin price analysis by providing accessible and accountable data sources, processing methods, and insights, fostering informed decision-making and trust in the Time Series Analysis project for Bitcoin Price Prediction using Prophet.	Data documentation tools. - Metadata management systems. - Data catalog solutions. - Access control mechanisms. - Data lineage and provenance tracking. - Data visualization tools. - Data governance frameworks. - Compliance and auditing tools.

THANK YOU